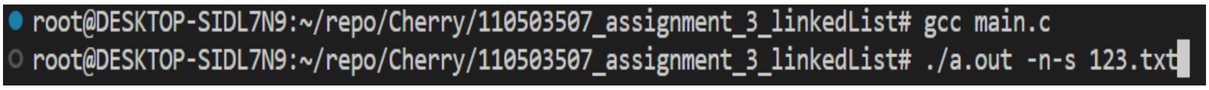
Assignment #3

110503507 林亞誼

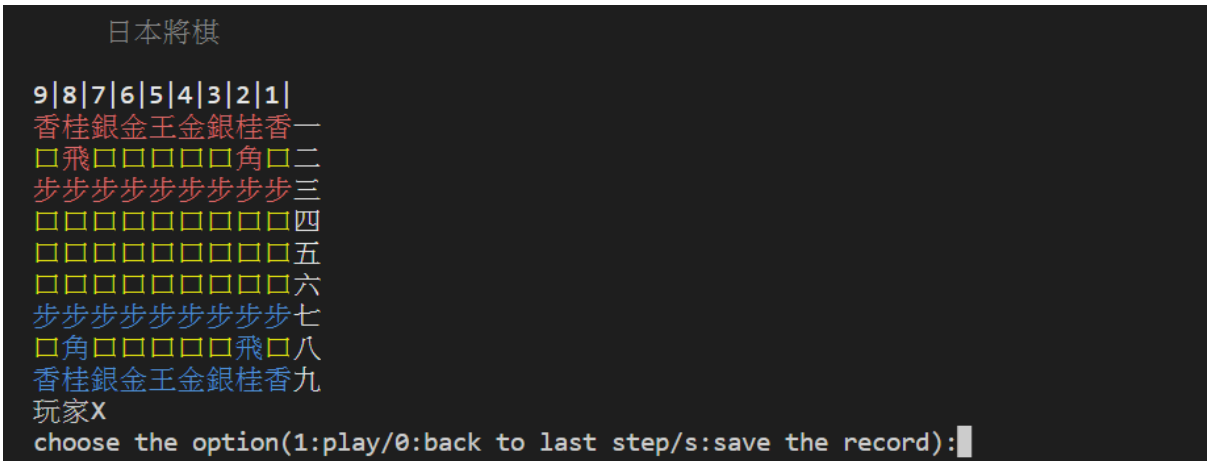
1. 編譯結果



```
root@DESKTOP-SIDL7N9:~/repo/Cherry/110503507_assignment_3_linkedList# gcc main.c
root@DESKTOP-SIDL7N9:~/repo/Cherry/110503507_assignment_3_linkedList# ./a.out -n-s 123.txt
```

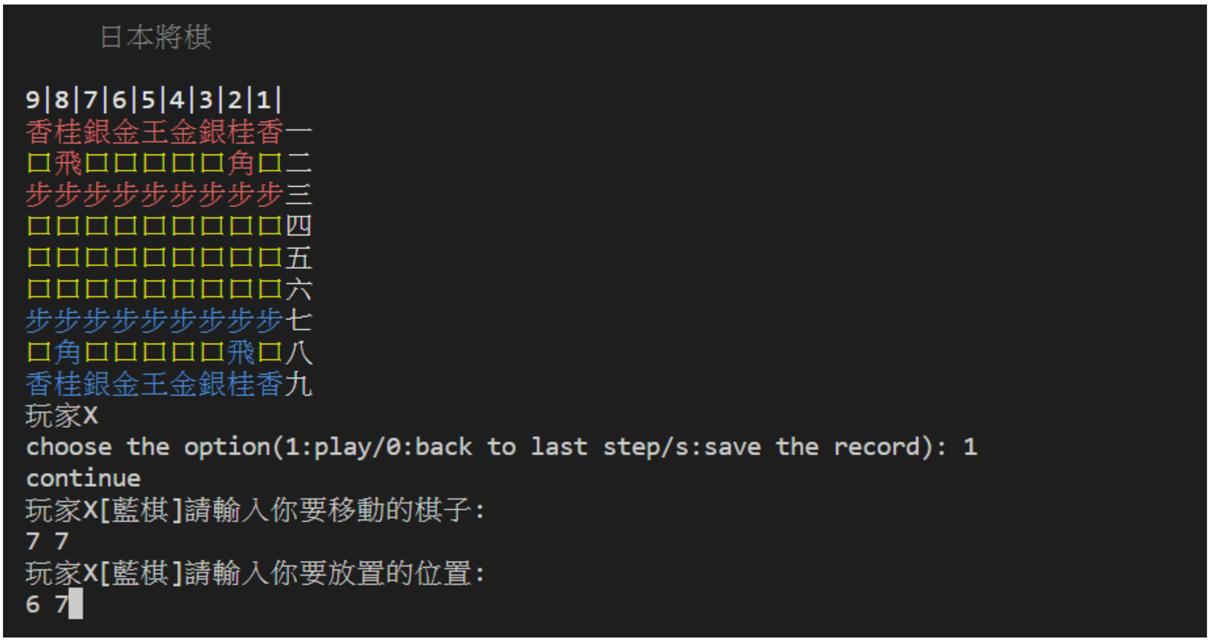
Figure 1: Compile file

2.執行結果



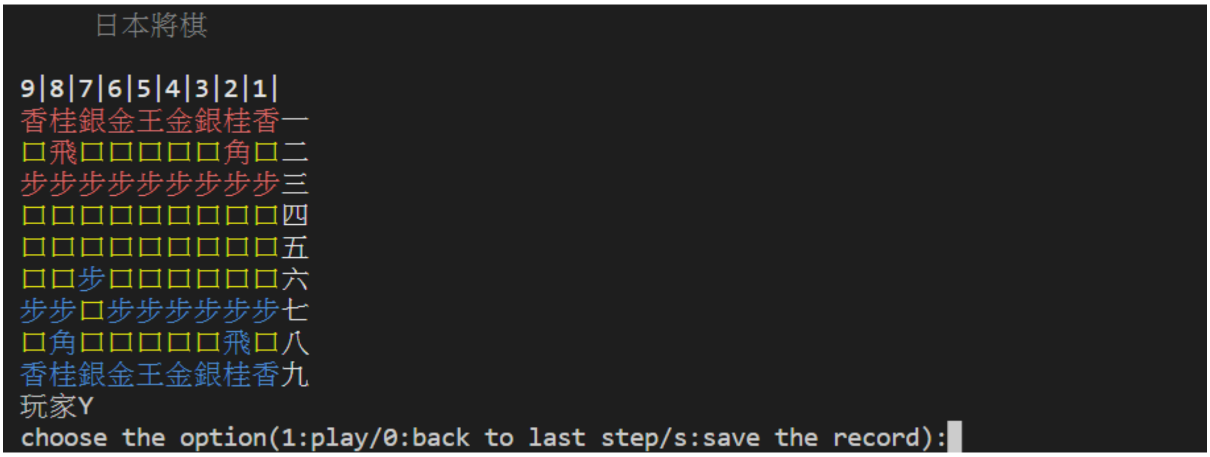
```
      日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口口口口口口口口六
步步步步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record):
```

Figure 2: Initial interface of X player



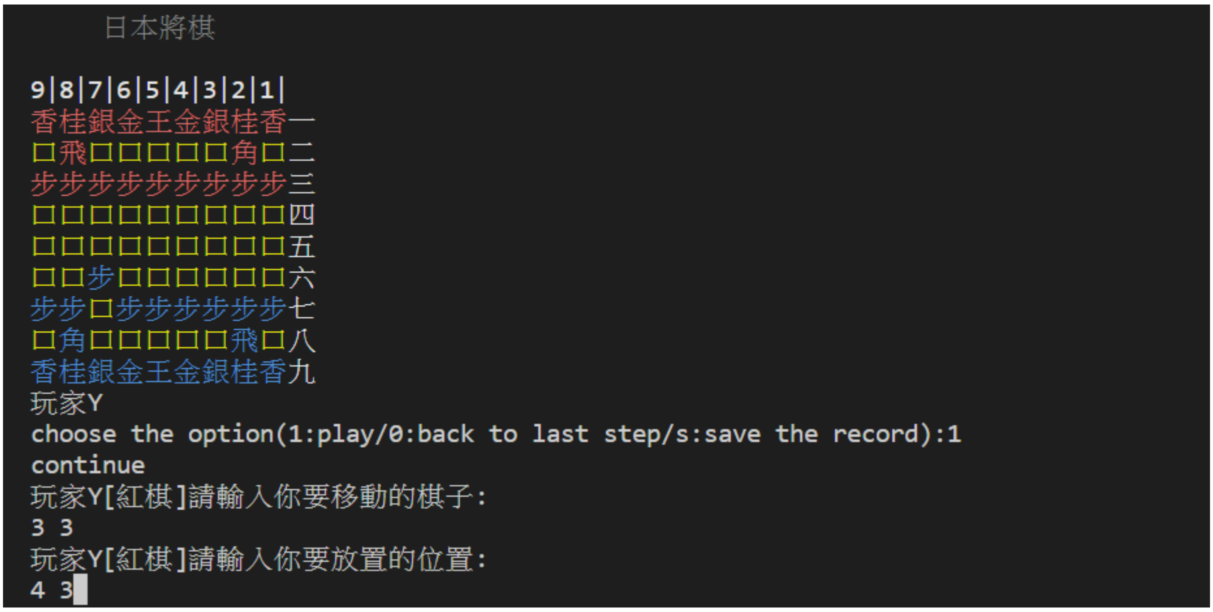
```
      日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口口口口口口口口六
步步步步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record): 1
continue
玩家X[藍棋]請輸入你要移動的棋子:
7 7
玩家X[藍棋]請輸入你要放置的位置:
6 7
```

Figure 3: X player inputs 1 to play and move the piece



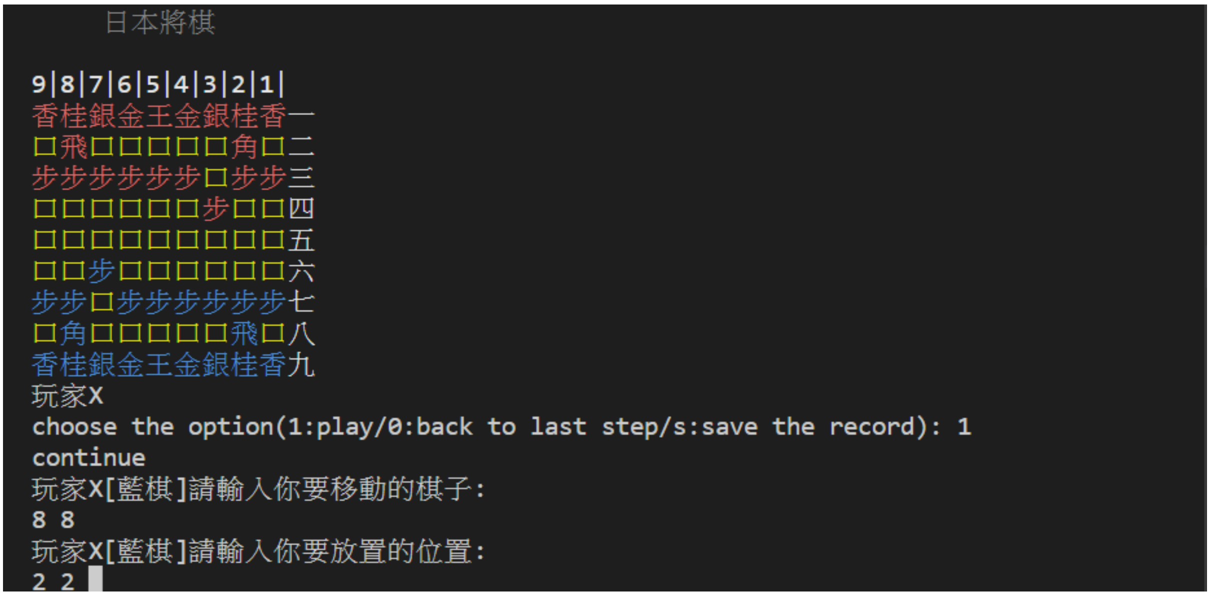
```
      日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record):
```

Figure 4: Initial interface of Y player



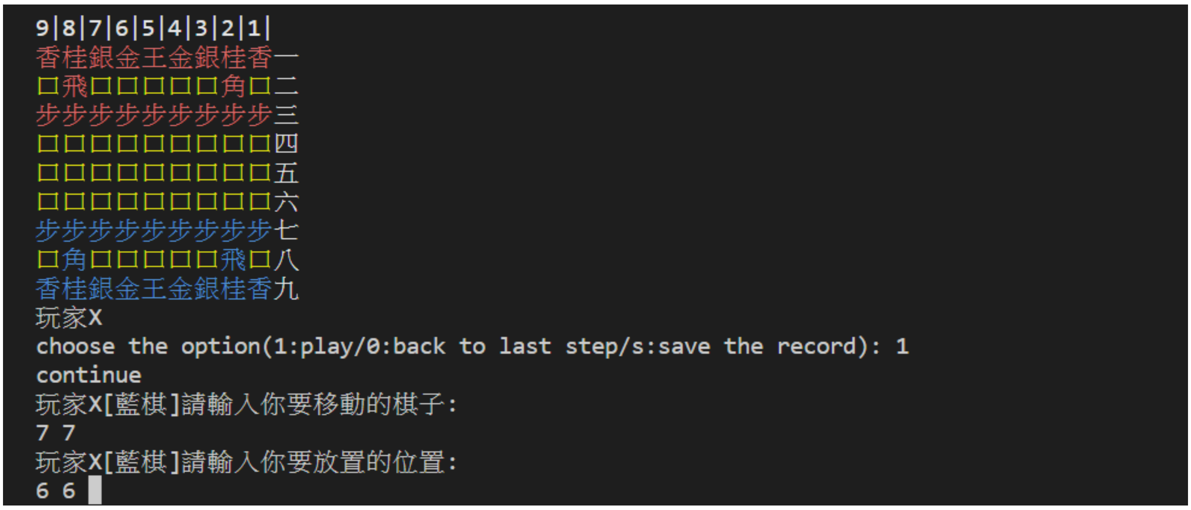
```
      日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record):1
continue
玩家Y[紅棋]請輸入你要移動的棋子:
3 3
玩家Y[紅棋]請輸入你要放置的位置:
4 3
```

Figure 5: Y player inputs 1 to play and move the piece



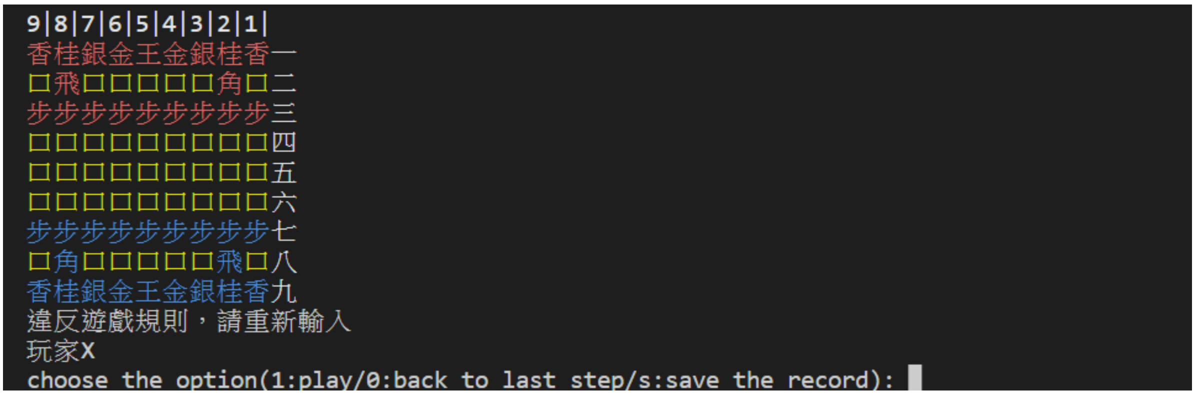
```
      日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record): 1
continue
玩家X[藍棋]請輸入你要移動的棋子:
8 8
玩家X[藍棋]請輸入你要放置的位置:
2 2
```

Figure 6: X player captures the piece of Y player



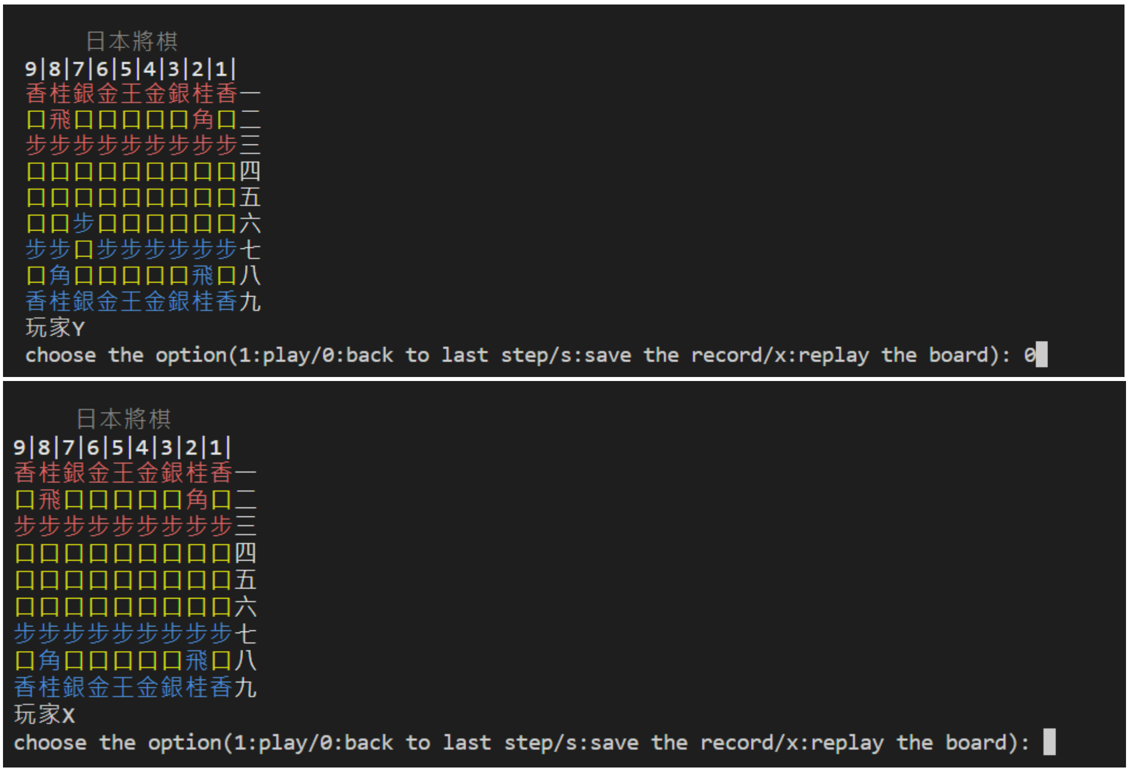
```
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口口口口口口口口六
步步步步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record): 1
continue
玩家X[藍棋]請輸入你要移動的棋子:
7 7
玩家X[藍棋]請輸入你要放置的位置:
6 6
```

Figure 7: when X player enters wrong value



```
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口口口口口口口口六
步步步步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
違反遊戲規則，請重新輸入
玩家X
choose the option(1:play/0:back to last step/s:save the record):
```

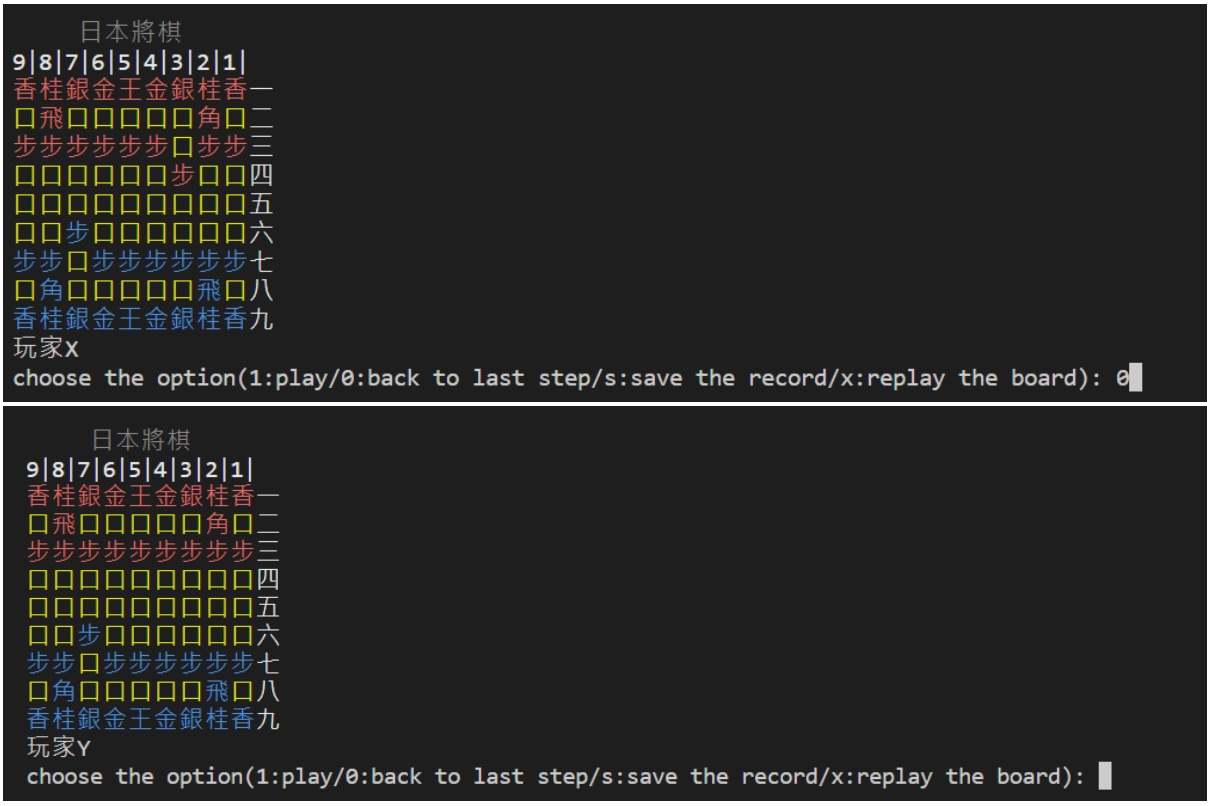
Figure 8: X player should input numbers again



```
    日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record/x:replay the board): 0
```

```
    日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口口口口口口口口六
步步步步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record/x:replay the board):
```

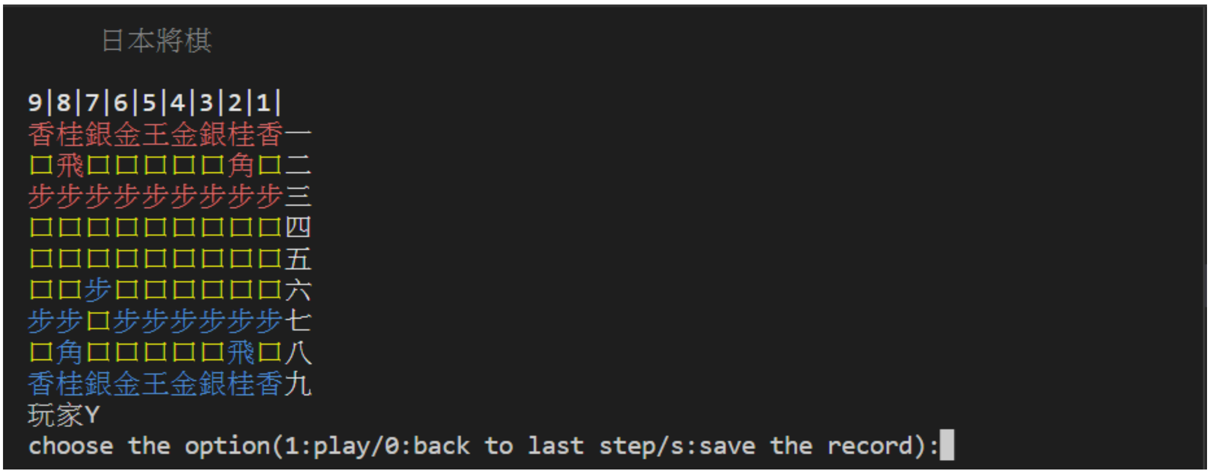
Figure 9: Y player enters 0 to go back to the last step



```
      日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record/x:replay the board): 0
```

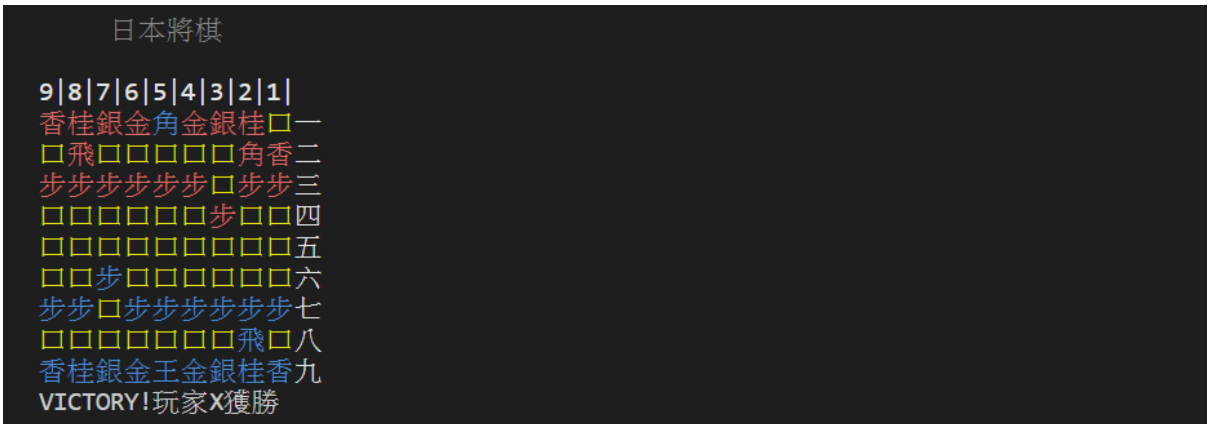
```
      日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record/x:replay the board):
```

Figure 10: X player enters 0 to go back to the last step



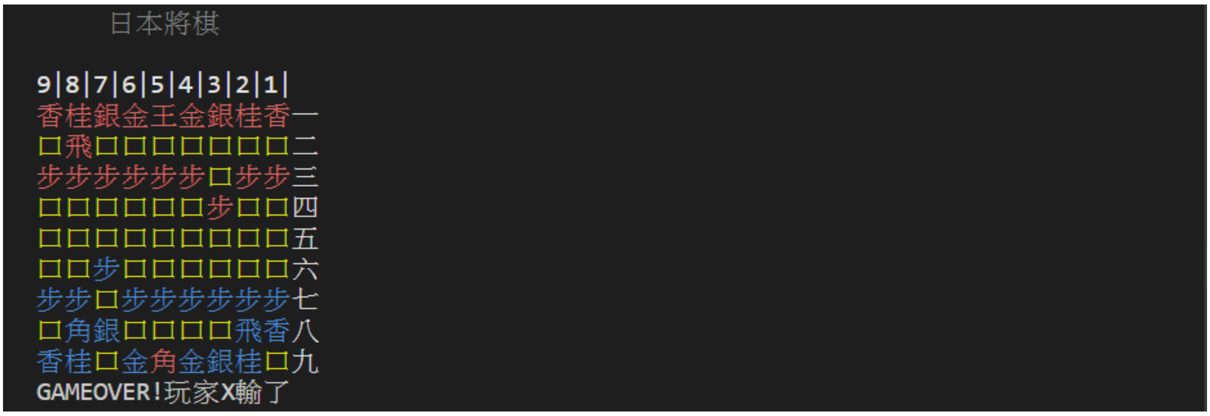
```
      日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record):
```

Figure 11: Pawn of Y player returns to initial place



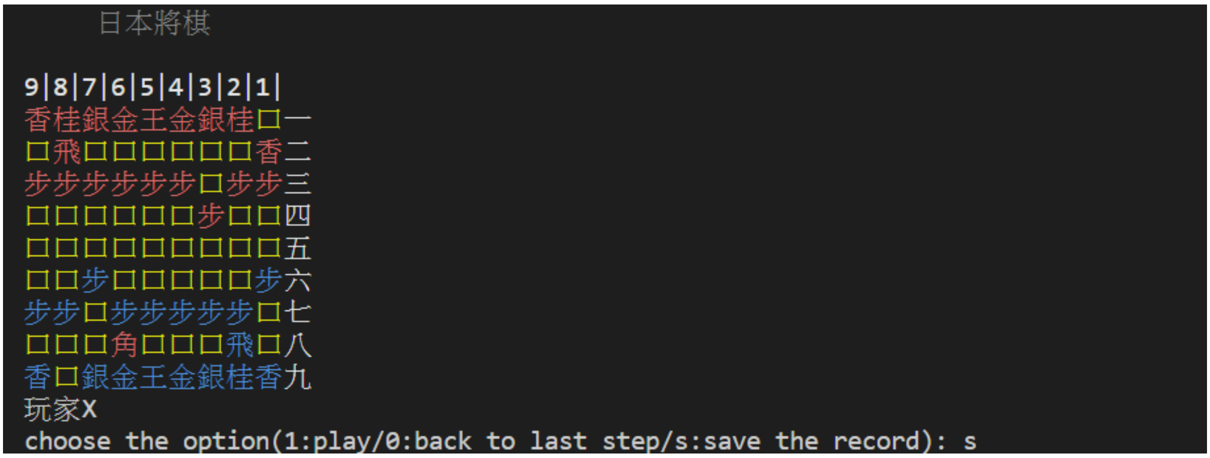
```
        日本將棋

 9|8|7|6|5|4|3|2|1|
 香桂銀金角金銀桂口一
 口飛口口口口口角香二
 步步步步步步口步步三
 口口口口口口步口口四
 口口口口口口口口口五
 口口步口口口口口口六
 步步口步步步步步步七
 口口口口口口口飛口八
 香桂銀金王金銀桂香九
 VICTORY!玩家X獲勝
```

Figure 12: X player wins the game



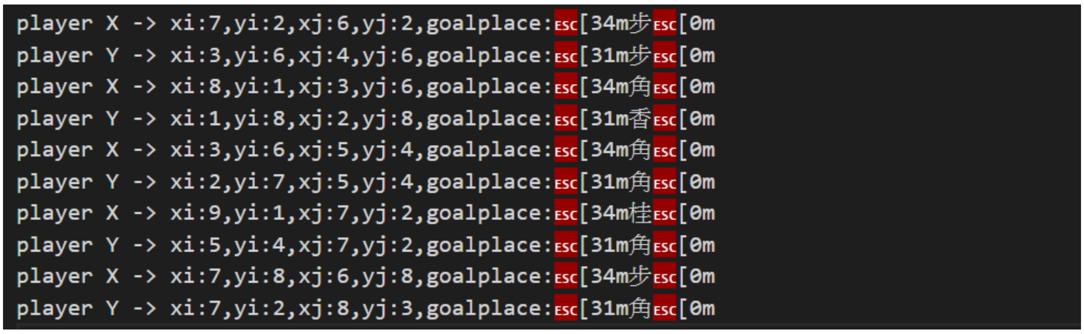
```
        日本將棋

 9|8|7|6|5|4|3|2|1|
 香桂銀金王金銀桂香一
 口飛口口口口口口口二
 步步步步步步口步步三
 口口口口口口步口口四
 口口口口口口口口口五
 口口步口口口口口口六
 步步口步步步步步步七
 口角銀口口口口飛香八
 香桂口金角金銀桂口九
 GAMEOVER!玩家X輸了
```

Figure 13: X player loses the game



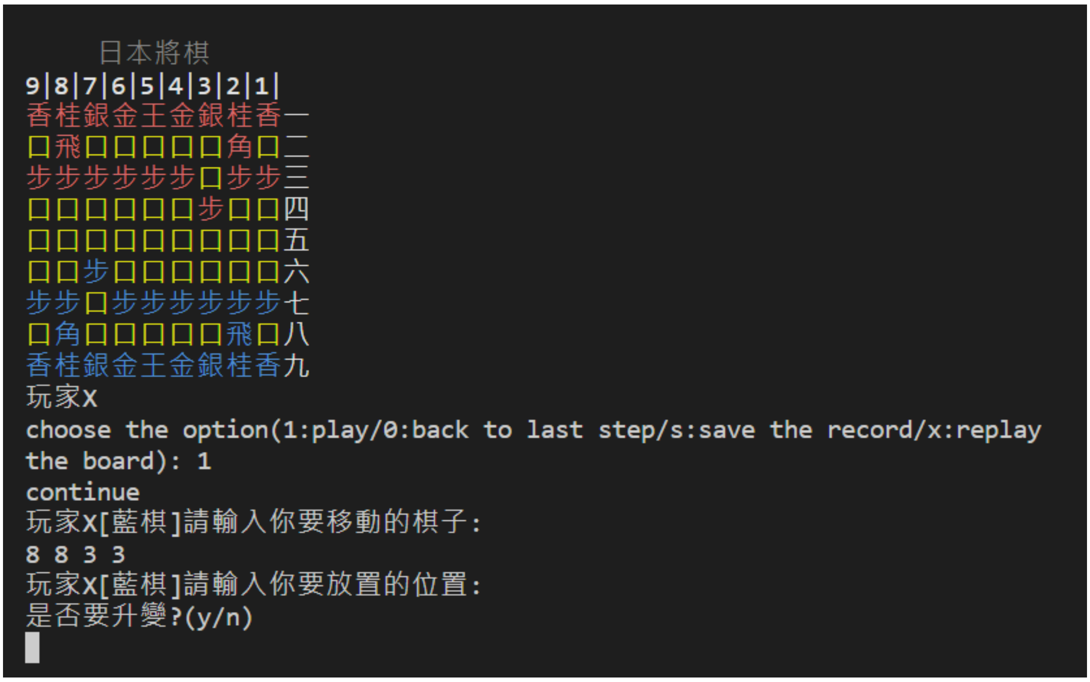
```
        日本將棋

 9|8|7|6|5|4|3|2|1|
 香桂銀金王金銀桂口一
 口飛口口口口口口香二
 步步步步步步口步步三
 口口口口口口步口口四
 口口口口口口口口口五
 口口步口口口口口步六
 步步口步步步步步口七
 口口口角口口口飛口八
 香口銀金王金銀桂香九
 玩家X
 choose the option(1:play/0:back to last step/s:save the record): s
```

Figure 14: Once we input s , the record would be saved



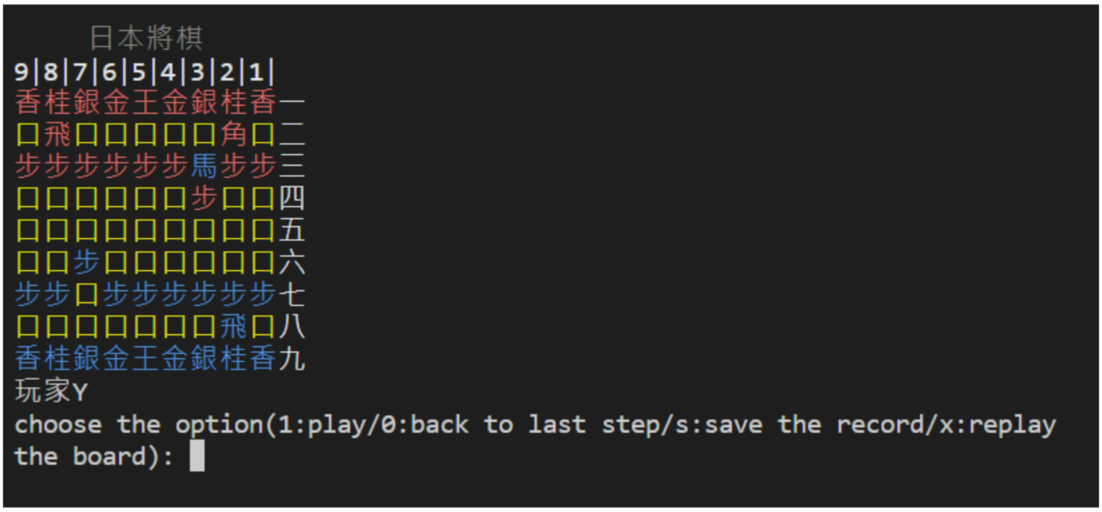
```
player X -> xi:7,yi:2,xj:6,yj:2,goalplace:ESC[34m步ESC[0m
player Y -> xi:3,yi:6,xj:4,yj:6,goalplace:ESC[31m步ESC[0m
player X -> xi:8,yi:1,xj:3,yj:6,goalplace:ESC[34m角ESC[0m
player Y -> xi:1,yi:8,xj:2,yj:8,goalplace:ESC[31m香ESC[0m
player X -> xi:3,yi:6,xj:5,yj:4,goalplace:ESC[34m角ESC[0m
player Y -> xi:2,yi:7,xj:5,yj:4,goalplace:ESC[31m角ESC[0m
player X -> xi:9,yi:1,xj:7,yj:2,goalplace:ESC[34m桂ESC[0m
player Y -> xi:5,yi:4,xj:7,yj:2,goalplace:ESC[31m角ESC[0m
player X -> xi:7,yi:8,xj:6,yj:8,goalplace:ESC[34m步ESC[0m
player Y -> xi:7,yi:2,xj:8,yj:3,goalplace:ESC[31m角ESC[0m
```

Figure 15: The record would be printed on [record.txt]



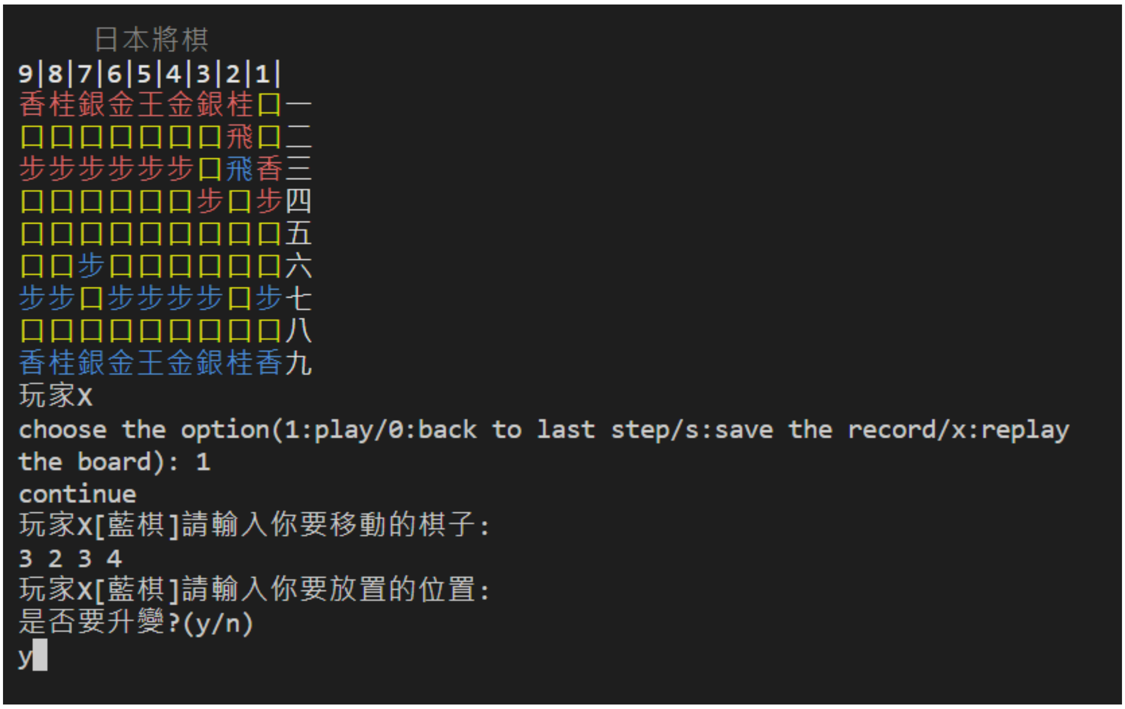
```
      日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record/x:replay
the board): 1
continue
玩家X[藍棋]請輸入你要移動的棋子:
8 8 3 3
玩家X[藍棋]請輸入你要放置的位置:
是否要升變?(y/n)
```

Figure 16: If input 'y', the shogi will be promotion



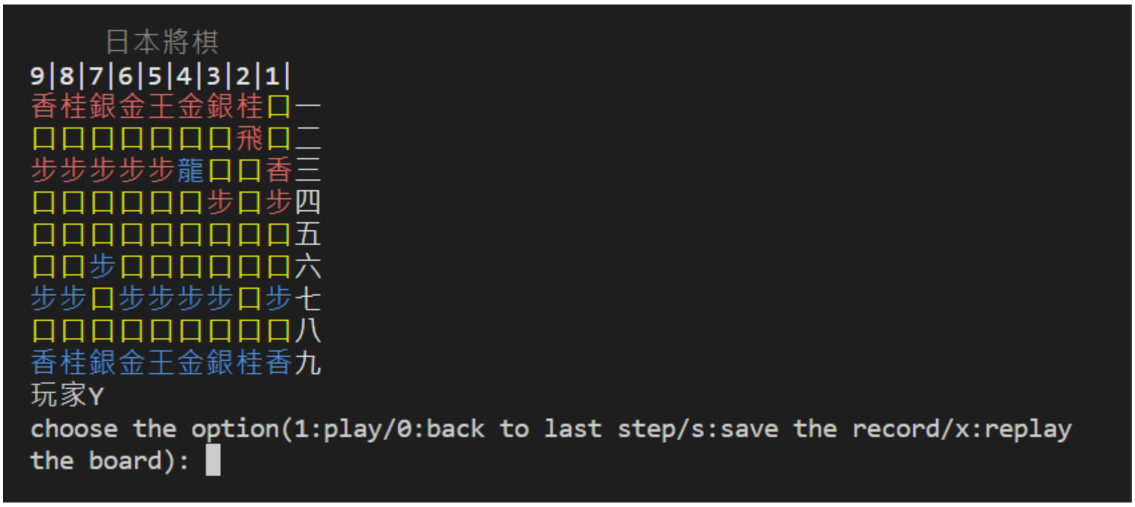
```
      日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步馬步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口口口口口口口飛口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record/x:replay
the board):
```

Figure 17: “龍馬” is the promotion of “角”



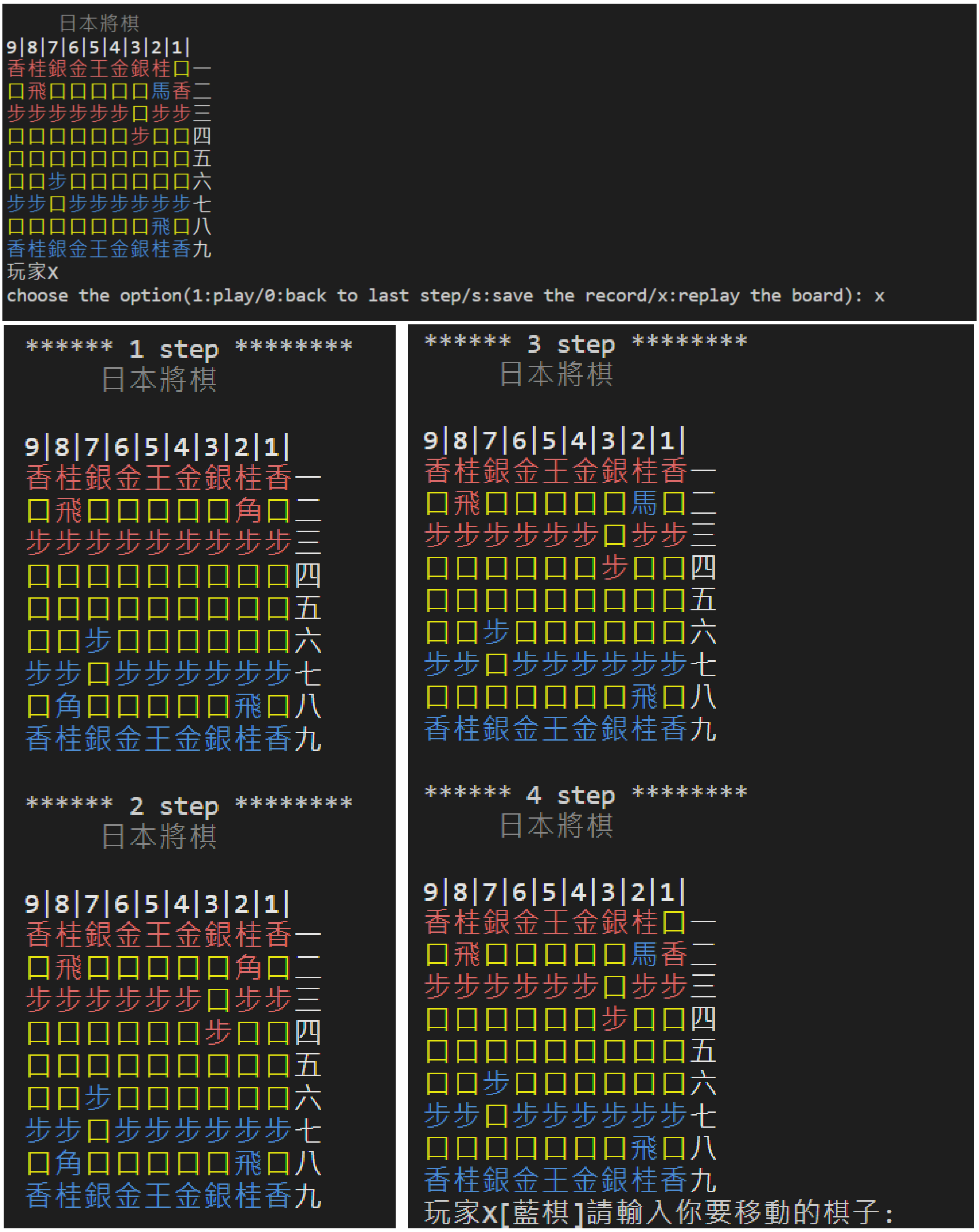
```
     日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂口一
口口口口口口口飛口二
步步步步步步口飛香三
口口口口口口步口步四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步口步七
口口口口口口口口口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record/x:replay
the board): 1
continue
玩家X[藍棋]請輸入你要移動的棋子:
3 2 3 4
玩家X[藍棋]請輸入你要放置的位置:
是否要升變?(y/n)
y
```

Figure 18: “If input 'y', the shogi will be promotion



```
     日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂口一
口口口口口口口飛口二
步步步步步龍口口香三
口口口口口口步口步四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步口步七
口口口口口口口口口八
香桂銀金王金銀桂香九
玩家Y
choose the option(1:play/0:back to last step/s:save the record/x:replay
the board): 
```

Figure 19: “龍王” is the promotion of “飛”



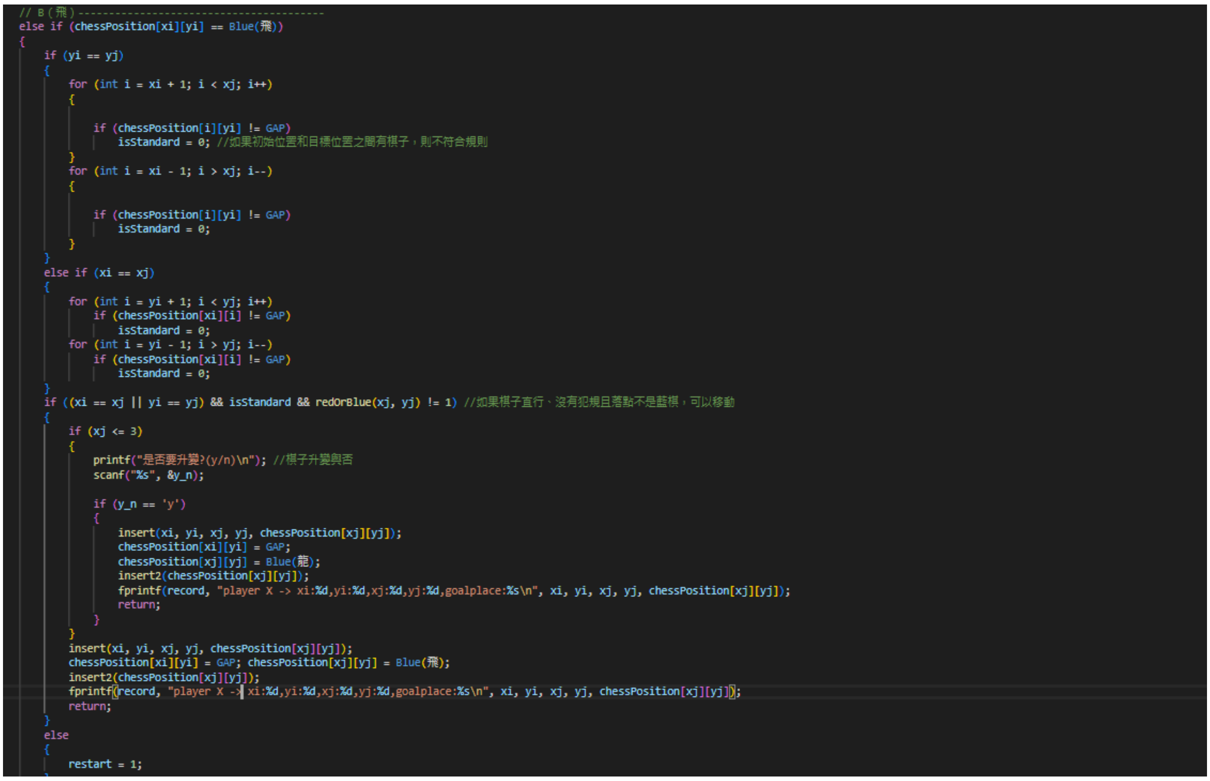
```
      日本將棋
9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂口一
口飛口口口口口馬香二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口口口口口口口飛口八
香桂銀金王金銀桂香九
玩家X
choose the option(1:play/0:back to last step/s:save the record/x:replay the board): x
```

```
****** 1 step ********
     日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步步步步三
口口口口口口口口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九

****** 2 step ********
     日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口角口二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口角口口口口口飛口八
香桂銀金王金銀桂香九
```

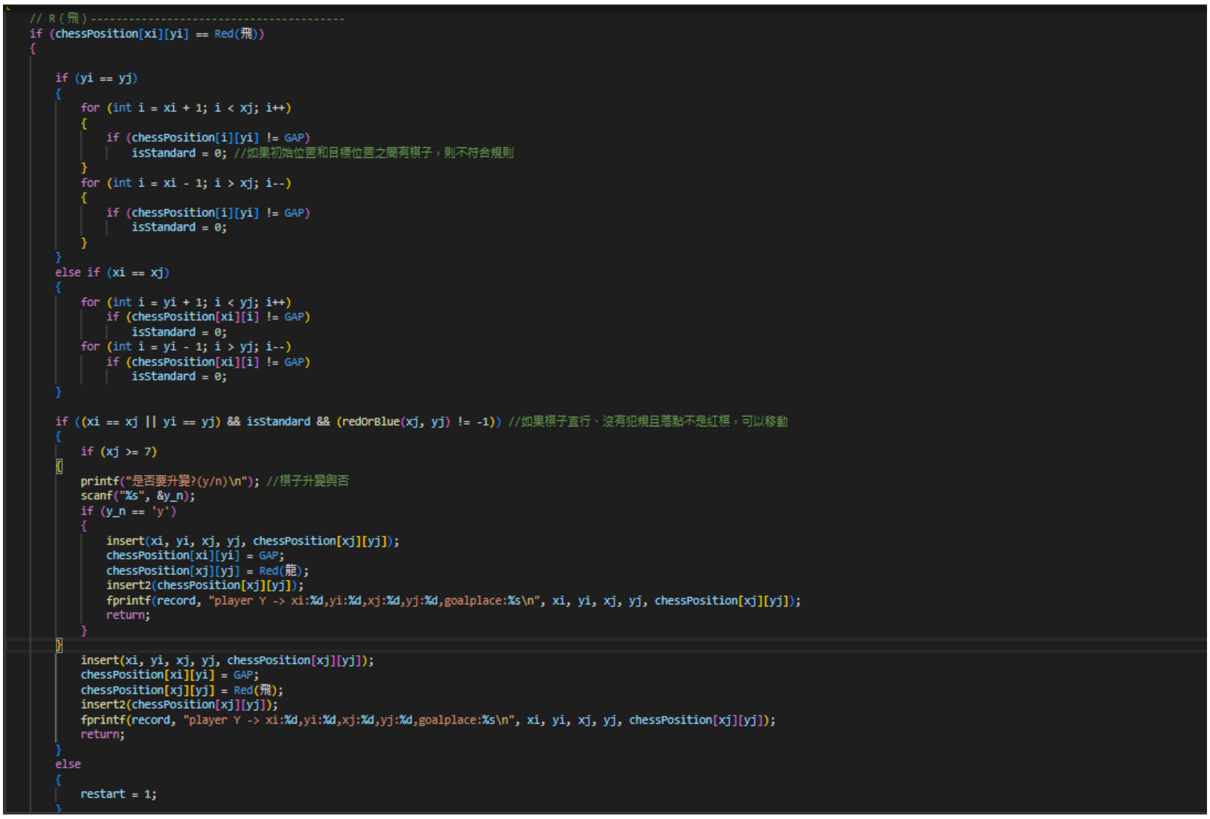
```
****** 3 step ********
     日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂香一
口飛口口口口口馬口二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口口口口口口口飛口八
香桂銀金王金銀桂香九

****** 4 step ********
     日本將棋

9|8|7|6|5|4|3|2|1|
香桂銀金王金銀桂口一
口飛口口口口口馬香二
步步步步步步口步步三
口口口口口口步口口四
口口口口口口口口口五
口口步口口口口口口六
步步口步步步步步步七
口口口口口口口飛口八
香桂銀金王金銀桂香九
玩家X[藍棋]請輸入你要移動的棋子:
```

Figure 20: “If input 'x', the shogi will replay

<將棋棋子規則>

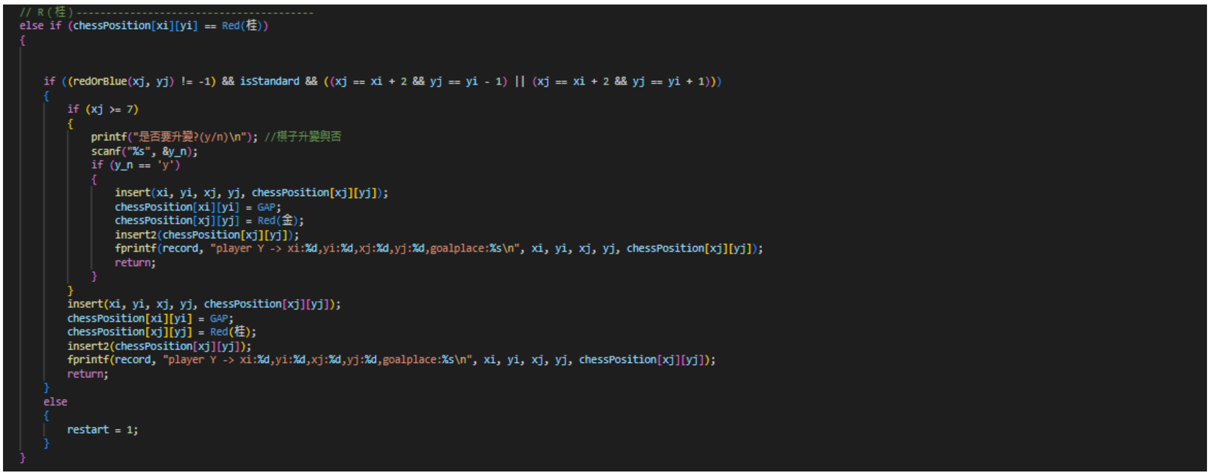


```c
// B (飛) --------------------------------------
else if (chessPosition[xi][yi] == Blue(飛))
{
    if (yi == yj)
    {
        for (int i = xi + 1; i < xj; i++)
        {

            if (chessPosition[i][yi] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        for (int i = xi - 1; i > xj; i--)
        {

            if (chessPosition[i][yi] != GAP)
                isStandard = 0;
        }
    }
    else if (xi == xj)
    {
        for (int i = yi + 1; i < yj; i++)
            if (chessPosition[xi][i] != GAP)
                isStandard = 0;
        for (int i = yi - 1; i > yj; i--)
            if (chessPosition[xi][i] != GAP)
                isStandard = 0;
    }
    if ((xi == xj || yi == yj) && isStandard && redOrBlue(xj, yj) != 1) //如果棋子直行、沒有犯規且落點不是藍棋，可以移動
    {
        if (xj <= 3)
        {
            printf("是否要升變?(y/n)\n"); //棋子升變與否
            scanf("%s", &y_n);

            if (y_n == 'y')
            {
                insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                chessPosition[xi][yi] = GAP;
                chessPosition[xj][yj] = Blue(龍);
                insert2(chessPosition[xj][yj]);
                fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                return;
            }
        }
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP; chessPosition[xj][yj] = Blue(飛);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        return;
    }
    else
    {
        restart = 1;
```



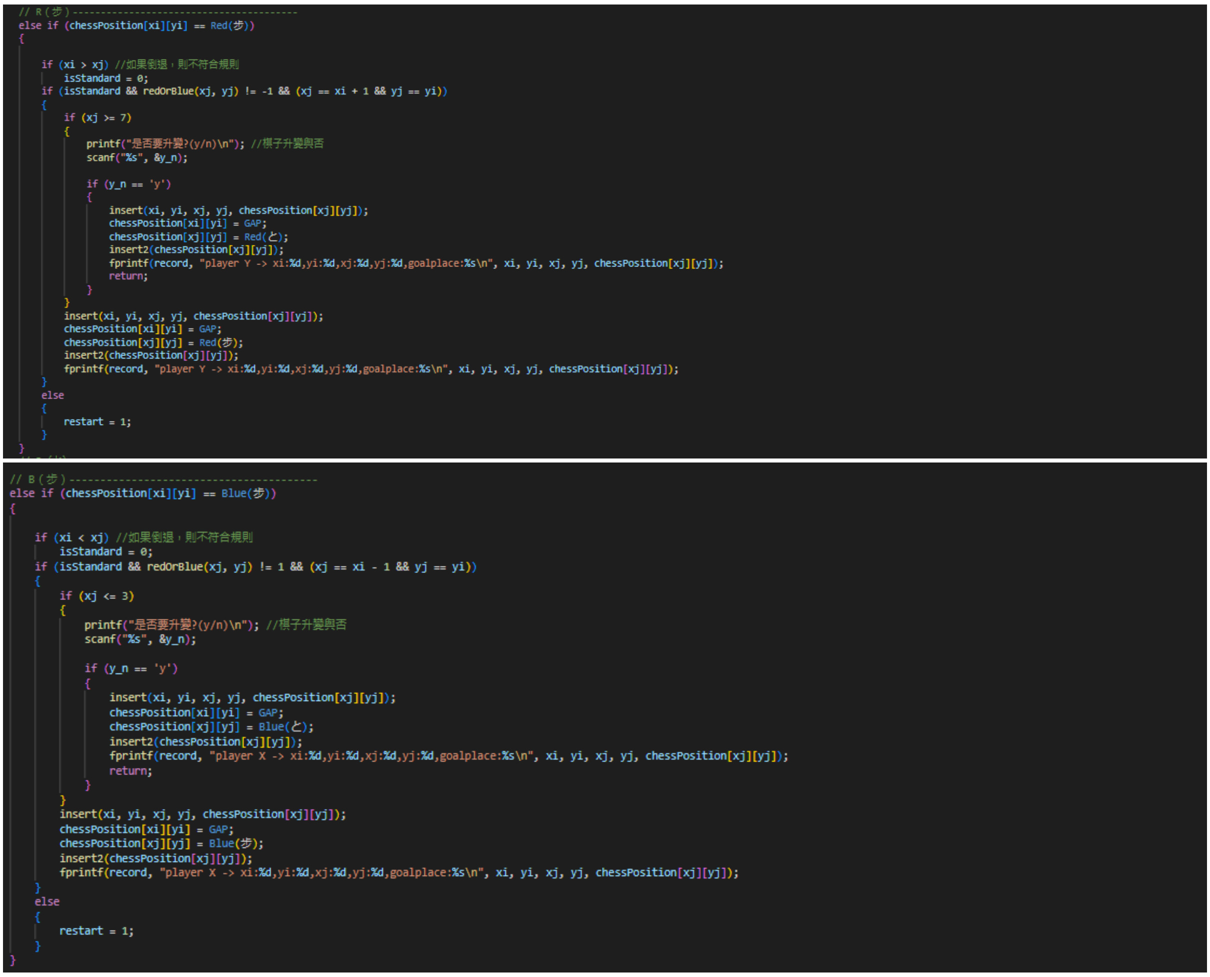
```c
// R (飛) ---------------------------------------
if (chessPosition[xi][yi] == Red(飛))
{

    if (yi == yj)
    {
        for (int i = xi + 1; i < xj; i++)
        {
            if (chessPosition[i][yi] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        for (int i = xi - 1; i > xj; i--)
        {
            if (chessPosition[i][yi] != GAP)
                isStandard = 0;
        }
    }
    else if (xi == xj)
    {
        for (int i = yi + 1; i < yj; i++)
            if (chessPosition[xi][i] != GAP)
                isStandard = 0;
        for (int i = yi - 1; i > yj; i--)
            if (chessPosition[xi][i] != GAP)
                isStandard = 0;
    }

    if ((xi == xj || yi == yj) && isStandard && (redOrBlue(xj, yj) != -1)) //如果棋子直行、沒有犯規且落點不是紅棋，可以移動
    {
        if (xj >= 7)
    {
        printf("是否要升變?(y/n)\n"); //棋子升變與否
        scanf("%s", &y_n);
        if (y_n == 'y')
        {
            insert(xi, yi, xj, yj, chessPosition[xj][yj]);
            chessPosition[xi][yi] = GAP;
            chessPosition[xj][yj] = Red(龍);
            insert2(chessPosition[xj][yj]);
            fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
            return;
        }
    }
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(飛);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        return;
    }
    else
    {
        restart = 1;
    }
```

Figure 21: [飛] 可到上下左右的任何 1 格



```c
// R (桂) ----------------------------------------
else if (chessPosition[xi][yi] == Red(桂))
{


    if ((redOrBlue(xj, yj) != -1) && isStandard && ((xj == xi + 2 && yj == yi - 1) || (xj == xi + 2 && yj == yi + 1)))
    {
        if (xj >= 7)
        {
            printf("是否要升變?(y/n)\n"); //棋子升變與否
            scanf("%s", &y_n);
            if (y_n == 'y')
            {
                insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                chessPosition[xi][yi] = GAP;
                chessPosition[xj][yj] = Red(金);
                insert2(chessPosition[xj][yj]);
                fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                return;
            }
        }
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(桂);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        return;
    }
    else
    {
        restart = 1;
    }
}
```

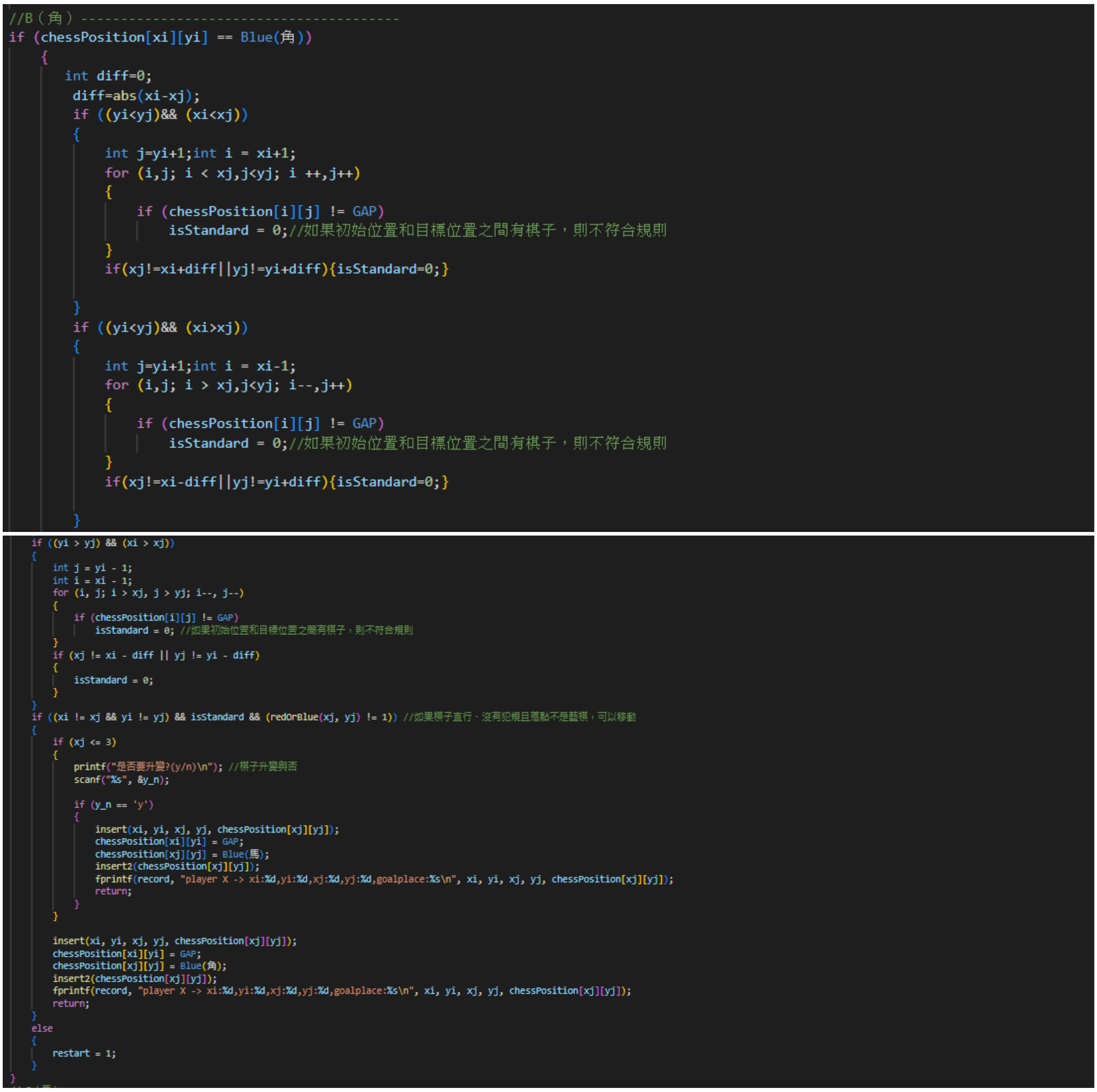
Figure 22: [桂] 每次走右上格或左上格對上之 1 格



```c
// R (步) ----------------------------------------
else if (chessPosition[xi][yi] == Red(步))
{

    if (xi > xj) //如果倒退，則不符合規則
        isStandard = 0;
    if (isStandard && redOrBlue(xj, yj) != -1 && (xj == xi + 1 && yj == yi))
    {
        if (xj >= 7)
        {
            printf("是否要升變?(y/n)\n"); //棋子升變與否
            scanf("%s", &y_n);

            if (y_n == 'y')
            {
                insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                chessPosition[xi][yi] = GAP;
                chessPosition[xj][yj] = Red(と);
                insert2(chessPosition[xj][yj]);
                fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                return;
            }
        }
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(步);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
    }
    else
    {
        restart = 1;
    }
}
```

```c
// B (步) ----------------------------------------
else if (chessPosition[xi][yi] == Blue(步))
{

    if (xi < xj) //如果倒退，則不符合規則
        isStandard = 0;
    if (isStandard && redOrBlue(xj, yj) != 1 && (xj == xi - 1 && yj == yi))
    {
        if (xj <= 3)
        {
            printf("是否要升變?(y/n)\n"); //棋子升變與否
            scanf("%s", &y_n);

            if (y_n == 'y')
            {
                insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                chessPosition[xi][yi] = GAP;
                chessPosition[xj][yj] = Blue(と);
                insert2(chessPosition[xj][yj]);
                fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                return;
            }
        }
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Blue(步);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
    }
    else
    {
        restart = 1;
    }
}
```

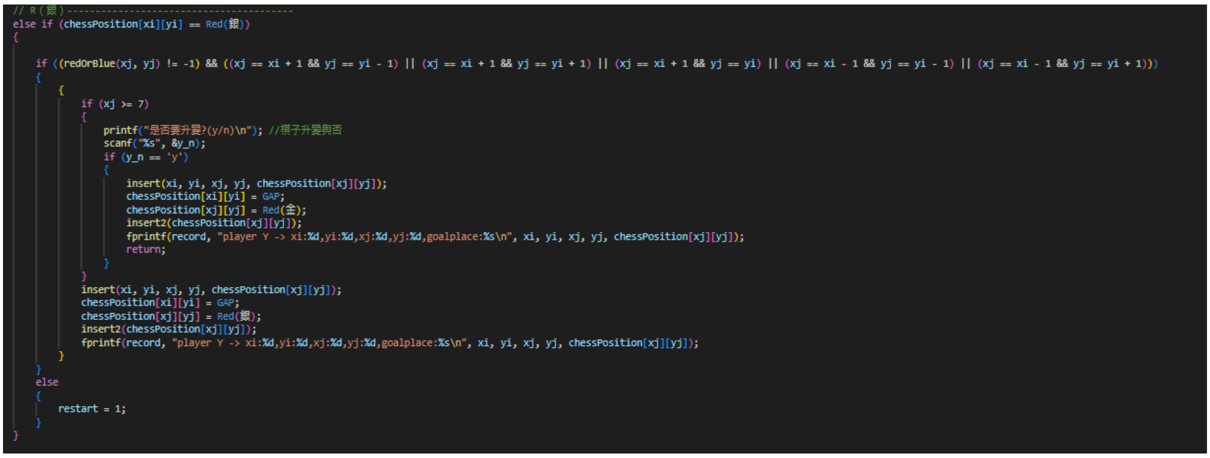
Figure 23: [步] 每次只可向前 1 格，不能後退



```
//B（角）----------------------------------------
if (chessPosition[xi][yi] == Blue(角))
    {
       int diff=0;
        diff=abs(xi-xj);
        if ((yi<yj)&& (xi<xj))
        {
            int j=yi+1;int i = xi+1;
            for (i,j; i < xj,j<yj; i ++,j++)
            {
                if (chessPosition[i][j] != GAP)
                    isStandard = 0;//如果初始位置和目標位置之間有棋子，則不符合規則
            }
            if(xj!=xi+diff||yj!=yi+diff){isStandard=0;}

        }
        if ((yi<yj)&& (xi>xj))
        {
            int j=yi+1;int i = xi-1;
            for (i,j; i > xj,j<yj; i--,j++)
            {
                if (chessPosition[i][j] != GAP)
                    isStandard = 0;//如果初始位置和目標位置之間有棋子，則不符合規則
            }
            if(xj!=xi-diff||yj!=yi+diff){isStandard=0;}

        }
```

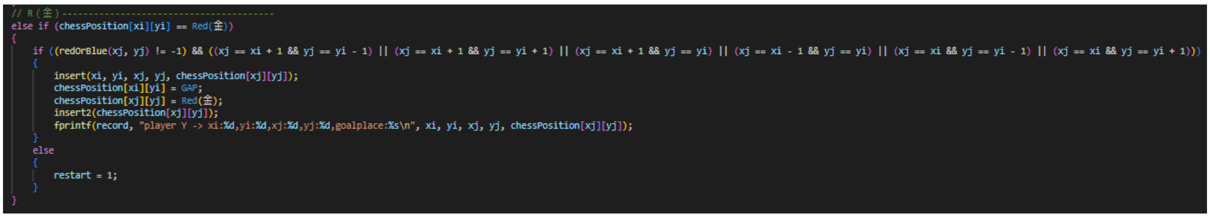
```
    if ((yi > yj) && (xi > xj))
    {
        int j = yi - 1;
        int i = xi - 1;
        for (i, j; i > xj, j > yj; i--, j--)
        {
            if (chessPosition[i][j] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        if (xj != xi - diff || yj != yi - diff)
        {
            isStandard = 0;
        }
    }
    if ((xi != xj && yi != yj) && isStandard && (redOrBlue(xj, yj) != 1)) //如果棋子直行、沒有犯規且落點不是藍棋，可以移動
    {
        if (xj <= 3)
        {
            printf("是否要升變?(y/n)\n"); //棋子升變與否
            scanf("%s", &y_n);

            if (y_n == 'y')
            {
                insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                chessPosition[xi][yi] = GAP;
                chessPosition[xj][yj] = Blue(馬);
                insert2(chessPosition[xj][yj]);
                fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                return;
            }
        }

        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Blue(角);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        return;
    }
    else
    {
        restart = 1;
    }
}
```

Figure 24: [角] 每次可到對角的任何 1 格(英文字母「X」方向)



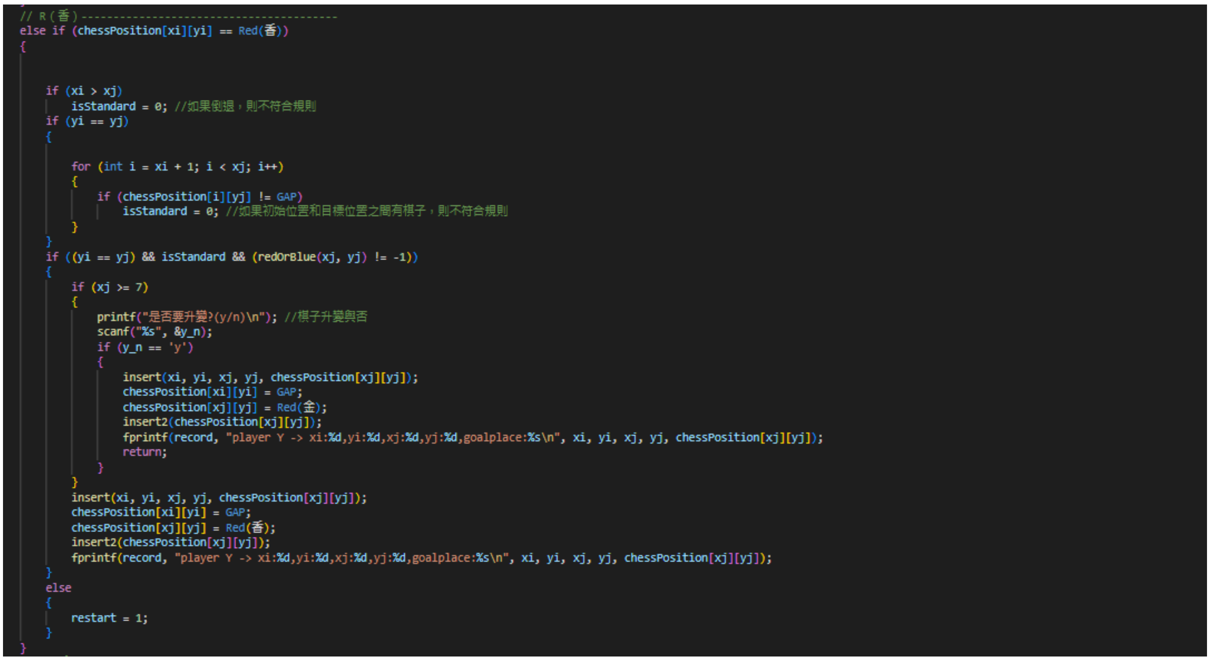
```
// R（銀）----------------------------------------
else if (chessPosition[xi][yi] == Red(銀))
{

    if ((redOrBlue(xj, yj) != -1) && ((xj == xi + 1 && yj == yi - 1) || (xj == xi + 1 && yj == yi + 1) || (xj == xi + 1 && yj == yi) || (xj == xi - 1 && yj == yi - 1) || (xj == xi - 1 && yj == yi + 1)))
    {
        {
            if (xj >= 7)
            {
                printf("是否要升變?(y/n)\n"); //棋子升變與否
                scanf("%s", &y_n);
                if (y_n == 'y')
                {
                    insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                    chessPosition[xi][yi] = GAP;
                    chessPosition[xj][yj] = Red(金);
                    insert2(chessPosition[xj][yj]);
                    fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                    return;
                }
            }
            insert(xi, yi, xj, yj, chessPosition[xj][yj]);
            chessPosition[xi][yi] = GAP;
            chessPosition[xj][yj] = Red(銀);
            insert2(chessPosition[xj][yj]);
            fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        }
    }
    else
    {
        restart = 1;
    }
}
```

Figure 25: [銀] 每次走前面、右上、右下、左上、左下 1 格



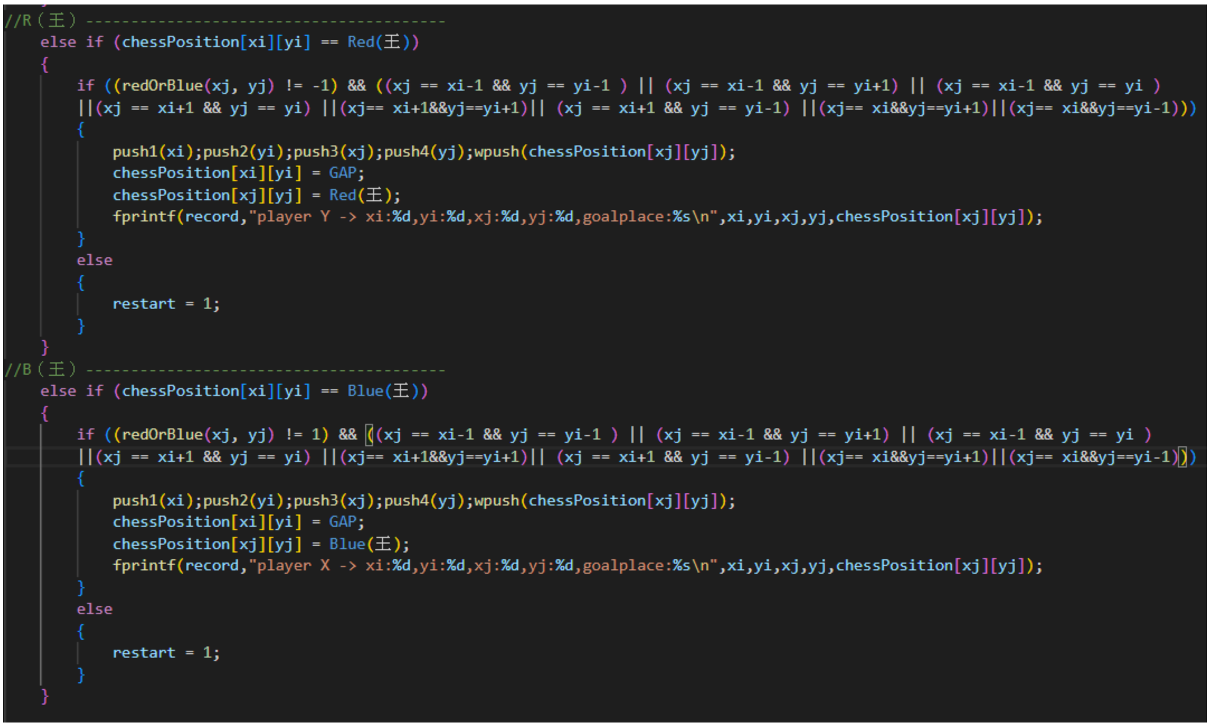
```
// R(金)---------------------------------------
else if (chessPosition[xi][yi] == Red(金))
{
    if ((redOrBlue(xj, yj) != -1) && ((xj == xi + 1 && yj == yi - 1) || (xj == xi + 1 && yj == yi + 1) || (xj == xi + 1 && yj == yi) || (xj == xi - 1 && yj == yi) || (xj == xi && yj == yi - 1) || (xj == xi && yj == yi + 1)))
    {
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(金);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
    }
    else
    {
        restart = 1;
    }
}
```

Figure 26: [金] 每次走前面、右上、右面、左上、左面、下面 1 格



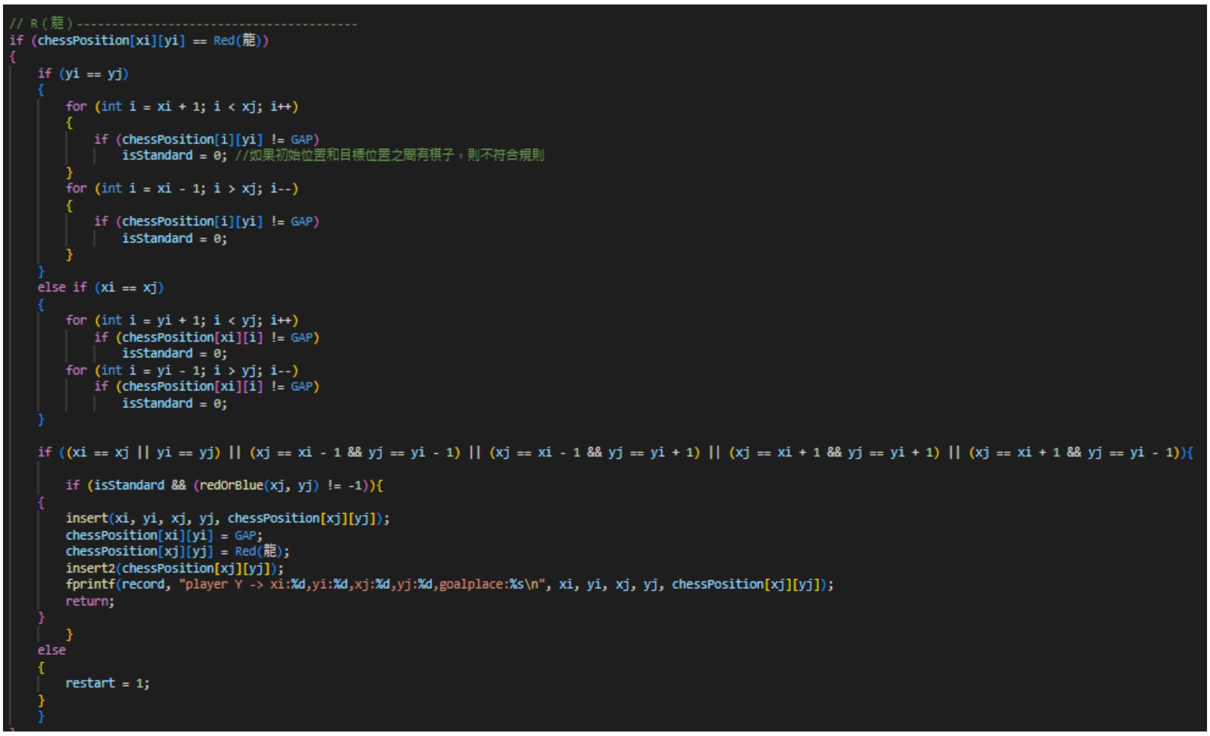
```
// R(香)---------------------------------------
else if (chessPosition[xi][yi] == Red(香))
{


    if (xi > xj)
        isStandard = 0; //如果倒退，則不符合規則
    if (yi == yj)
    {

        for (int i = xi + 1; i < xj; i++)
        {
            if (chessPosition[i][yj] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
    }
    if ((yi == yj) && isStandard && (redOrBlue(xj, yj) != -1))
    {
        if (xj >= 7)
        {
            printf("是否要升變?(y/n)\n"); //棋子升變與否
            scanf("%s", &y_n);
            if (y_n == 'y')
            {
                insert(xi, yi, xj, yj, chessPosition[xj][yj]);
                chessPosition[xi][yi] = GAP;
                chessPosition[xj][yj] = Red(金);
                insert2(chessPosition[xj][yj]);
                fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
                return;
            }
        }
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(香);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
    }
    else
    {
        restart = 1;
    }
}
```

Figure 27: [香] 每次可向前行任 1 格，但不能後退



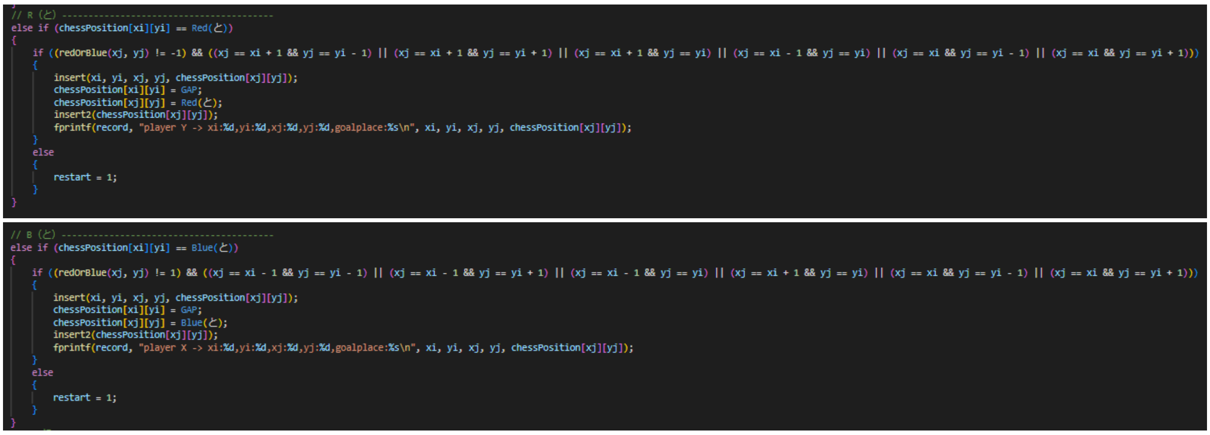
```
//R(王) ---------------------------------------
    else if (chessPosition[xi][yi] == Red(王))
    {
        if ((redOrBlue(xj, yj) != -1) && ((xj == xi-1 && yj == yi-1 ) || (xj == xi-1 && yj == yi+1) || (xj == xi-1 && yj == yi )
        ||(xj == xi+1 && yj == yi) ||(xj== xi+1&&yj==yi+1)|| (xj == xi+1 && yj == yi-1) ||(xj== xi&&yj==yi+1)||(xj== xi&&yj==yi-1)))
        {
            push1(xi);push2(yi);push3(xj);push4(yj);wpush(chessPosition[xj][yj]);
            chessPosition[xi][yi] = GAP;
            chessPosition[xj][yj] = Red(王);
            fprintf(record,"player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n",xi,yi,xj,yj,chessPosition[xj][yj]);
        }
        else
        {
            restart = 1;
        }
    }
//B(王) ---------------------------------------
    else if (chessPosition[xi][yi] == Blue(王))
    {
        if ((redOrBlue(xj, yj) != 1) && ((xj == xi-1 && yj == yi-1 ) || (xj == xi-1 && yj == yi+1) || (xj == xi-1 && yj == yi )
        ||(xj == xi+1 && yj == yi) ||(xj== xi+1&&yj==yi+1)|| (xj == xi+1 && yj == yi-1) ||(xj== xi&&yj==yi+1)||(xj== xi&&yj==yi-1)))
        {
            push1(xi);push2(yi);push3(xj);push4(yj);wpush(chessPosition[xj][yj]);
            chessPosition[xi][yi] = GAP;
            chessPosition[xj][yj] = Blue(王);
            fprintf(record,"player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n",xi,yi,xj,yj,chessPosition[xj][yj]);
        }
        else
        {
            restart = 1;
        }
    }
```

Figure 28: [王] 向前面、右上、右面、右下、左上、左面、左下、下面 1 格



```
// R (龍) ----------------------------------------
if (chessPosition[xi][yi] == Red(龍))
{
    if (yi == yj)
    {
        for (int i = xi + 1; i < xj; i++)
        {
            if (chessPosition[i][yi] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        for (int i = xi - 1; i > xj; i--)
        {
            if (chessPosition[i][yi] != GAP)
                isStandard = 0;
        }
    }
    else if (xi == xj)
    {
        for (int i = yi + 1; i < yj; i++)
            if (chessPosition[xi][i] != GAP)
                isStandard = 0;
        for (int i = yi - 1; i > yj; i--)
            if (chessPosition[xi][i] != GAP)
                isStandard = 0;
    }

    if ((xi == xj || yi == yj) || (xj == xi - 1 && yj == yi - 1) || (xj == xi - 1 && yj == yi + 1) || (xj == xi + 1 && yj == yi + 1) || (xj == xi + 1 && yj == yi - 1)){

        if (isStandard && (redOrBlue(xj, yj) != -1)){
    {
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(龍);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        return;
    }
        }
    else
    {
        restart = 1;
    }
    }
```

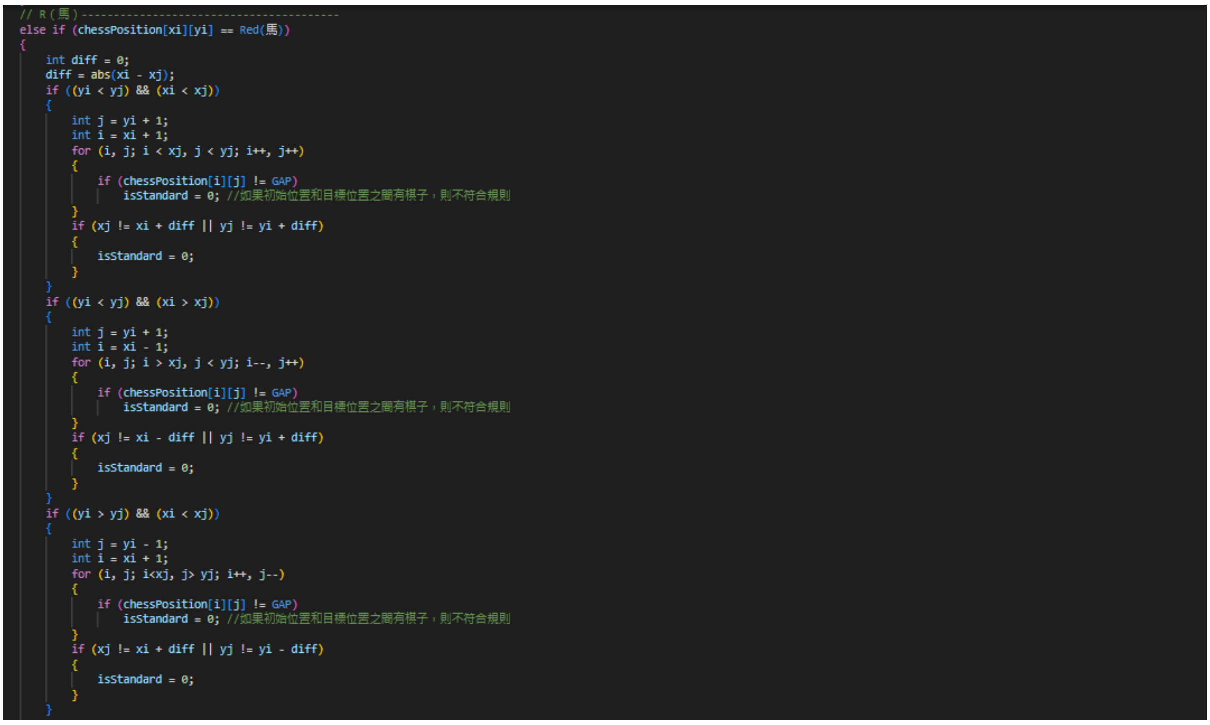
Figure 29: [龍] 向前面、右上、右面、右下、左上、左面、左下、下面 1 格、向同一行或同一列移動



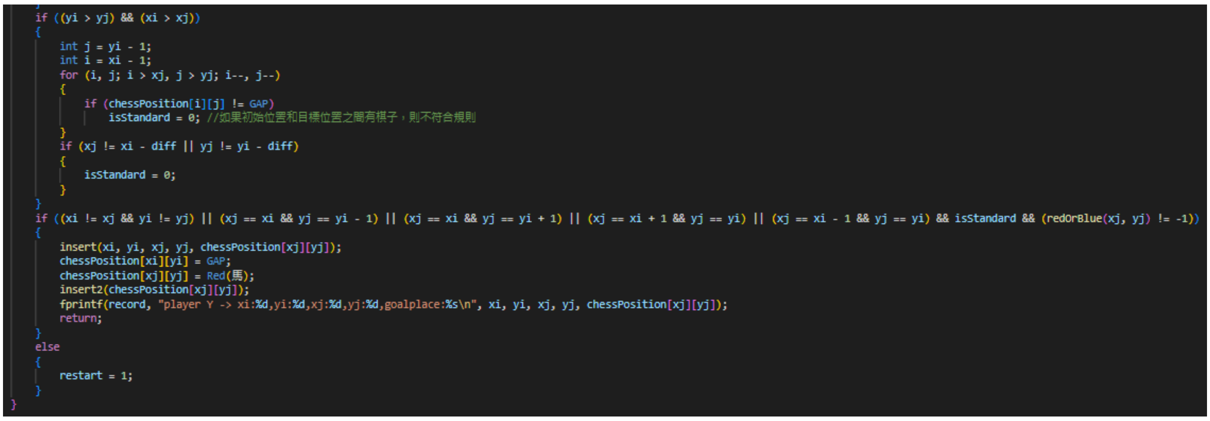
```
// R (と) ----------------------------------------
else if (chessPosition[xi][yi] == Red(と))
{
    if ((redOrBlue(xj, yj) != -1) && ((xj == xi + 1 && yj == yi - 1) || (xj == xi + 1 && yj == yi + 1) || (xj == xi + 1 && yj == yi) || (xj == xi - 1 && yj == yi) || (xj == xi && yj == yi - 1) || (xj == xi && yj == yi + 1)))
    {
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(と);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
    }
    else
    {
        restart = 1;
    }
}
```

```
// B (と) ----------------------------------------
else if (chessPosition[xi][yi] == Blue(と))
{
    if ((redOrBlue(xj, yj) != 1) && ((xj == xi - 1 && yj == yi - 1) || (xj == xi - 1 && yj == yi + 1) || (xj == xi - 1 && yj == yi) || (xj == xi + 1 && yj == yi) || (xj == xi && yj == yi - 1) || (xj == xi && yj == yi + 1)))
    {
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Blue(と);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player X -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
    }
    else
    {
        restart = 1;
    }
}
```

Figure 30: [と] 每次走前面、右上、右面、左上、左面、下面 1 格



```
// R(馬)----------------------------------------
else if (chessPosition[xi][yi] == Red(馬))
{
    int diff = 0;
    diff = abs(xi - xj);
    if ((yi < yj) && (xi < xj))
    {
        int j = yi + 1;
        int i = xi + 1;
        for (i, j; i < xj, j < yj; i++, j++)
        {
            if (chessPosition[i][j] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        if (xj != xi + diff || yj != yi + diff)
        {
            isStandard = 0;
        }
    }
    if ((yi < yj) && (xi > xj))
    {
        int j = yi + 1;
        int i = xi - 1;
        for (i, j; i > xj, j < yj; i--, j++)
        {
            if (chessPosition[i][j] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        if (xj != xi - diff || yj != yi + diff)
        {
            isStandard = 0;
        }
    }
    if ((yi > yj) && (xi < xj))
    {
        int j = yi - 1;
        int i = xi + 1;
        for (i, j; i<xj, j> yj; i++, j--)
        {
            if (chessPosition[i][j] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        if (xj != xi + diff || yj != yi - diff)
        {
            isStandard = 0;
        }
    }
```



```
    if ((yi > yj) && (xi > xj))
    {
        int j = yi - 1;
        int i = xi - 1;
        for (i, j; i > xj, j > yj; i--, j--)
        {
            if (chessPosition[i][j] != GAP)
                isStandard = 0; //如果初始位置和目標位置之間有棋子，則不符合規則
        }
        if (xj != xi - diff || yj != yi - diff)
        {
            isStandard = 0;
        }
    }
    if ((xi != xj && yi != yj) || (xj == xi && yj == yi - 1) || (xj == xi && yj == yi + 1) || (xj == xi + 1 && yj == yi) || (xj == xi - 1 && yj == yi) && isStandard && (redOrBlue(xj, yj) != -1))
    {
        insert(xi, yi, xj, yj, chessPosition[xj][yj]);
        chessPosition[xi][yi] = GAP;
        chessPosition[xj][yj] = Red(馬);
        insert2(chessPosition[xj][yj]);
        fprintf(record, "player Y -> xi:%d,yi:%d,xj:%d,yj:%d,goalplace:%s\n", xi, yi, xj, yj, chessPosition[xj][yj]);
        return;
    }
    else
    {
        restart = 1;
    }
}
```

Figure 31: [馬] 向前面、右上、右面、右下、左上、左面、左下、下面 1 格、即向對角線位置移動

3.規則

當使用者輸入 1，可以決定下一步的位置，(先輸入[段(行)]的數值，再輸入[筋(列)]的數值) 確定要移動的棋子後，在決定放置的位置。如果輸入錯誤，會顯示違反遊戲規則，使用者需要再輸入一次(1/0/s/x) 。 若使用者輸入 0，可進行悔棋。悔棋可連續執行，直到回到第一手。若使用者輸入 x，可進行重播。當一方進入另外一方前三排的領地，除了王將 (玉將) 、金將及已經升級的棋子外，所有棋子都可以選擇是否升變。輸入 s 會儲存從頭到此的下棋資料到 record.txt。若其中一方將對方的王吃掉，遊戲即結束。

4.參考資料

(1) https://markdown.tw/

(2) https://shogi.hk/Gameplay-of-Japanese-Chess-Shogi/

(3) https://lakesd6531.pixnet.net/blog/post/332858496-%5B%E8%B3%87%E6%96%99%E7%B5%90%E6%A7%8B%5D%E7%94%A8c%E8%AA%9E%E8%A8%80%E8%A3%BD%E4%BD%9C%E5%A0%86%E7%96%8A%28stack%29

(4) https://www.delftstack.com/zh-tw/howto/c/read-file-c/

5. GITHUB 連結

https://github.com/NCU-DSA-111-1/assignment_2-yayi1213/tree/main/110503507_assignment_3_linked%20list